

[0040] What is claimed is:

## CLAIMS

1. A method for arranging a priority queue, comprising the following steps:

counting latency and cycle of each of the plurality of access requests in the priority queue;

counting total cycles of all access requests in the priority queue; and

comparing latency of each of the plurality of access requests with the total cycles;

wherein if a latency of an access request is smaller than the total cycles, then arranging the nonsatisfied access request into the priority queue once again, the access request is called nonsatisfied access request in which the latency is smaller than total cycles of all access requests in priority queue.

2. As the method of claim 1, wherein if the latency of each access request in the priority queue is larger than the total cycles, executing the plurality of access requests according to the sequence of the priority queue.

3. As the method of claim 1, further comprising:

counting the maximum during cycles between two same access requests and comparing with said latency when said nonsatisfied access request is arranged, if said latency is smaller than said maximum during cycles, then arranging said

access request into said priority queue again.

4. As the method of claim 1, further comprising:

recounting new total cycles when said nonsatisfied access request is arranged, and comparing with said latency of each access request, if the latency of another access request is smaller than said new total cycles, then arranging said access request into the priority once again, else executing the plurality of access requests according to the sequence of the priority queue.

5. As the method of claim 1, wherein the positions of additional arranged access requests are the positions which can let interval of each during cycle of these same access requests equally.

6. As the method of claim 1, wherein the method can be used in an arbitratative mechanism.

7. A method of arbitration for different priority levels, comprising the following steps:

receiving a plurality of access requests;

separating the plurality of access requests into a plurality of different priority levels; and

arranging the plurality of access requests of each priority level as a priority queue;

wherein the priority level of the plurality access requests can be dynamically adjusted according to the different requirements of the system.

8. As the method of claim 7, wherein the steps of arranging the priority queue of each priority level comprising:

counting each cycle and latency of a plurality of access requests;

counting total cycles of all access requests; and

comparing latency of each of the plurality of access requests with the total cycles;

wherein if a latency of an access request is smaller than the total cycles, then arranging the nonsatisfied access request into the priority queue once again, else executing the plurality of access requests according to the sequence of the priority queue, the access request is called nonsatisfied access request in which the latency is smaller than total cycles of all access requests in priority queue.

9. As the method of claim 8, further comprising:

counting the maximum during cycles between two same access requests and comparing with said latency when said nonsatisfied access request is arranged, if said latency is smaller than said maximum during cycles, then arranging said access request into said priority queue again.

10. As the method of claim 8, further comprising:

recounting new total cycles when said nonsatisfied access request is arranged, and comparing with said each latency of each access request, if the latency of another access request is smaller than said new total cycles, then arranging said access request into the priority once again, else executing the plurality of access requests according to the sequence of the priority queue.

11. As the method of claim 8, wherein the positions of additional arranged access requests are the positions which can let interval of each during cycle of those same access requests equally.

12. As the method of claim 7, wherein each access request is respectively arranged in priority queue with corresponding request number.

13. As the method of claim 7, further comprising:

selecting one of the access requests of one of the priority queue to execute, and pointing out the next access request which has the priority.

14. As the method of claim 7, after finishing executing all access requests of a priority level in priority queue, then executing one of the access request of next priority level, then go back to execute the access request of highest priority level.

15. An arbitrate apparatus of access request arbitration, comprising:

a plurality of access request selectors, wherein each one of the plurality of access request selectors receiving a plurality of access requests respectively; and

an ownership selector, coupled to the plurality of access request selectors, when an access request is being executed, an asking point out signal is sent out to ask for pointing out the position of next access request;

wherein the arbitrate apparatus has its own priority level, and each access request selectors respectively selects one access requests grouped to respected priority level from the

plurality of access requests, and said ownership selector receives the plurality of access requests respectively and arranges the plurality of access requests which are selected by the plurality of access request selectors into a priority queue.

16. As the arbitratative apparatus of claim 15, further comprising:

a priority setting register coupled to the plurality of access request selectors for setting request numbers of access requests that belong to said priority level.

17. As the arbitratative apparatus of claim 16, further comprising:

an ownership multiplexer, finding the corresponding request number of access request from register according to the position of said priority queue.

18. As the arbitratative apparatus of claim 15, wherein said ownership selector further comprises a next ownership selector unit, pointing out the position of the next access request when receiving said asking point out signal.

19. As the arbitratative apparatus of claim 15, further comprising:

an OR gate with multi-inputs coupled between the plurality of access request selectors and said ownership selector and an output coupled to the ownership selector of one higher priority level, estimating whether there any access request is asked of this priority level, if there is, then arranging the access request into the last position of the priority queue of one higher priority level.

20. As the arbitratative apparatus of claim 15, further comprising:

a 2-input AND gate, in which one input coupled to said

ownership selector, estimating if said asking point out signal is sent, another input coupled to a estimation signal, estimating if the next access request is at the last position of the priority queue, and an output coupled to the ownership selector of one lower priority level, when both inputs are true, then pointing out the next access request of the one lower priority level.

21. As the arbitratve apparatus of claim 15, at least one said arbitratve apparatus with different priority level can be combined as an arbitratve mechanism wherein said arbitratve mechanism can be an arbiter.

22. An arbiter of access request arbitration, comprising:

- a plurality of arbitratve apparatus, each one has its own priority level;

- wherein each one of the arbitratve apparatus with different priority level at least comprises:

- a plurality of access request selectors, wherein each one of the plurality of access request selectors receiving a plurality of access request respectively; and

- an ownership selector, coupled to the plurality of access request selectors, when an access request is being executed, an asking point out signal is sent out to ask for pointing out the position of next access request;

- wherein each access request selectors respectively selects one access requests grouped to the priority level from the plurality of access requests, and said ownership selector

receives the plurality of access requests respectively and arranges the plurality of access requests which are selected by the plurality of access request selectors into a priority queue.

23. As an arbiter of claim 22, further comprising:

a priority setting register coupled to the plurality of access request selectors for setting request numbers of access requests that belong to said priority level.

24. As an arbiter of claim 23, further comprising:

an ownership multiplexer, finding the corresponding request number of access request from register according to the position of said priority queue.

25. As an arbiter of claim 22, wherein said ownership selector further comprises a next ownership selector unit, pointing out the position of the next access request when receiving said asking point out signal.

26. As an arbiter of claim 22, further comprising:

an OR gate with multi-inputs coupled between the plurality of access request selectors and said ownership selector and an output coupled to the ownership selector of one higher priority level, estimating whether there any access request is asked of this priority level, if there is, then arranging the access request into the priority queue of one higher priority level.

27. As an arbiter of claim 22, further comprising:

a 2-input AND gate, one input coupled to said ownership selector, estimating if said asking point out signal is sent, another input coupled to a estimation signal, estimating if the

next access request is the last access request of priority queue, and an output coupled to the ownership selector of one lower priority level, when both inputs are true, the one lower priority level point out the next access request.